بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

International Islamic University Chittagong

Department of Computer Science &Engineering (CSE)

Project Report

Course Code	CSE-2424
Course Title	Database Management System
Credit Hour	1.5
Semester	Autumn 2025

Submitted By		**Submitted To**	
Name of Students	**Team Leader :** Mohammad Sadman Tahiat **Members :** **1.** Minhaj Hasan Rohan **2.** Ahsan Miran	Name of Teacher	Mohammad Mubinur Rahman
Section	4CM	Designation	Lecturer
Semester	4th	Department	CSE
Group		Date of Final Submission	

Remark

Signature of the Teacher

Abstract:

The Emergency Contact System is a web-based platform developed to provide fast, reliable and location-specific access to essential emergency services including ambulances, hospitals, doctors, blood banks, blood donors, police stations, fire service stations and emergency shelters.
The system allows users to select their area and instantly view available emergency resources with contact details, availability status and other important information. An integrated admin panel enables authorized personnel to manage all service records through full CRUD (Create, Read, Update, Delete) operations.
The project was implemented using modern web technologies: HTML5, CSS3, JavaScript for the frontend, Node.js + Express.js for the backend, and MySQL as the relational database. The platform aims to reduce response time during critical situations by centralizing scattered emergency information into one accessible digital solution.

Motivation :

In today's fast-paced world, emergencies do not wait. A road accident, sudden cardiac arrest, fire outbreak, missing person, or natural disaster can strike at any moment. In those critical seconds, the difference between life and death often depends on how quickly the right help can be reached.
Unfortunately, in many parts of Bangladesh — especially in major cities like Chattogram and Dhaka — people still depend on scattered and unreliable methods to find emergency assistance. These include searching through personal mobile contacts, asking neighbors, relying on word-of-mouth, checking outdated social media posts, or making frantic phone calls to wrong or unavailable numbers. This outdated and chaotic approach frequently leads to confusion, significant delays, incorrect services being contacted, and in the worst cases, tragic and preventable loss of life.
The core motivation behind developing the **Emergency Contact System** was to completely eliminate this uncertainty and chaos. We wanted to create a single, centralized, reliable, and instantly accessible digital platform where any citizen — at any time of the day or night — can find the nearest and most appropriate emergency resource exactly according to their current location. Whether it is an ambulance, hospital, blood donor with a specific blood group, police station, fire service, or emergency shelter, the system filters and displays all relevant information within seconds.
We were deeply inspired by numerous real-life stories shared by families who lost precious time desperately searching for an ambulance during medical emergencies, or by individuals who struggled helplessly to locate a nearby blood donor with the required blood group during life-threatening situations. These heartbreaking real-world pain points strongly motivated us to design a practical solution that puts life-saving information directly into the hands of ordinary people — fast, accurate, and location-specific.
At the same time, we recognized that emergency service providers — ambulance operators, hospitals, blood banks, police stations, fire services, and others — also face significant

challenges. It is difficult for them to make their availability, updated contact details, and current status widely and reliably known to the public. Our platform bridges this critical gap by providing an easy and structured registration system for service providers, combined with a powerful, secure admin panel that allows administrators to verify, update, and maintain all information in real time, ensuring the database remains accurate and trustworthy.
Beyond addressing this urgent social and humanitarian problem, the project also gave our team an exciting opportunity to work hands-on with modern full-stack web technologies. From creating a clean and responsive frontend interface, to building secure and efficient RESTful APIs, to designing a well-structured relational database — every aspect of the system was thoughtfully planned to deliver high functionality, excellent performance, and a smooth user experience for both citizens and administrators.
Ultimately, our deepest motivation was to build something truly meaningful: a system that can genuinely contribute to faster emergency response times, help reduce preventable deaths, and bring a greater sense of safety, preparedness, and trust to communities across the country. We strongly believe that in moments of crisis, no one should ever feel helpless — help should always be just one click away.
This project represents our small but sincere effort toward making our society safer, smarter, more connected, and better prepared to face the most critical moments of life.

1. Introduction :

The Emergency Service Management System is a web-based application designed to provide quick and reliable access to essential emergency services such as ambulances, hospitals, blood banks, blood donors, police stations, fire service stations, and emergency shelters. The system allows users to search for nearby emergency resources based on their location, enabling faster response during critical situations. By centralizing emergency service information into a single digital platform, the project aims to improve accessibility, efficiency, and coordination between service providers and the public.
In many areas, especially during emergencies, people face significant difficulties in finding accurate and timely information about available emergency services. Traditionally, emergency contact details are scattered across different sources, such as phone directories, personal contacts, or local knowledge, which may not always be reliable or up to date. This lack of a centralized system often results in delays, confusion, and inefficient emergency response, which can put lives at risk.
The importance of this project lies in its ability to minimize response time and provide reliable information when it is needed the most. By offering location-based search, real-time availability details, and an admin-controlled management system, the Emergency Service Management System enhances public safety and supports better emergency decision-making. The system not only benefits users seeking immediate help but also assists administrators in efficiently managing and maintaining emergency service data, making it a valuable solution for modern emergency management.

2. Team Members :

The development of the **Emergency Contact System** was successfully carried out by a dedicated and skilled team of three members, each bringing unique strengths and expertise to ensure the creation of a robust, reliable, and user-friendly platform.

Team Leader :	Mohammad Sadman Tahiat (ID: C241100)
Team Member 1 :	Minhaj Hasan Rohan (ID: C241101)
Team Member 2 :	Ahsan Miran (ID: C241099)

Together, the team combined strong backend expertise, efficient database management, and modern frontend development skills to deliver a complete full-stack solution. Through regular collaboration, code reviews, testing, and iterative improvements, we successfully built a system that addresses real-world emergency needs while maintaining high standards of functionality, security, and usability.

Submission Link in Github :
https://github.com/Mohammad-Sadman/Project-Database-Management-System

3. Objectives :

1. To develop a centralized emergency service information system
2. To allow location-based searching of emergency services
3. To provide real-time availability information
4. To simplify emergency resource management
5. To ensure fast and reliable access to life-saving services

4. Technologies Used

The Emergency Contact System was developed using a modern, well-balanced full-stack technology stack that ensures responsiveness, performance, security, and scalability. Each technology was carefully selected to meet the specific requirements of a location-based emergency service platform.

Frontend Development :

HTML5- Used as the foundational markup language to structure all web pages, including the homepage, service category pages, registration forms, and admin dashboard. HTML5

provides semantic elements, improved accessibility, and native support for multimedia and responsive design features.
CSS3- Responsible for the complete visual styling and layout of the platform. Custom CSS was written to create a clean, modern, and professional user interface. Advanced features such as flexbox, grid layout, media queries, animations, gradients, and box shadows were utilized to achieve a fully responsive design that works seamlessly across desktops, tablets, and mobile devices.

Vanilla JavaScript (ES6+)- Implemented for all client-side interactivity and dynamic behavior. JavaScript handles real-time form validation, dynamic content loading using the Fetch API, asynchronous communication with the backend, session management for admin protection, event handling (clicks, form submissions), and updating the DOM without page reloads. No heavy frameworks were used, keeping the frontend lightweight, fast, and maintainable.

Backend Development :

Node.js- Served as the server-side runtime environment. Node.js enables fast, event-driven, non-blocking I/O operations, making it ideal for building a high-performance RESTful API that can handle multiple concurrent requests from users searching for emergency services.

Express.js- Used as the lightweight and flexible web application framework built on Node.js. Express.js powers the entire backend logic, including routing, middleware handling, request/response management, RESTful API endpoint creation (GET, POST, PUT, DELETE), error handling, and JSON data parsing. It provides a clean structure for organizing routes for different emergency services (ambulance, hospital, blood bank, etc.) and admin functionalities.

Database Management :

MySQL- Chosen as the relational database management system to store and manage all emergency service data efficiently. MySQL offers excellent performance for structured data, strong support for complex queries (area-based filtering, joins), referential integrity, indexing for fast searches, and scalability for future growth. All tables (ambulance, hospital, doctor, bloodbank, personinformation, police_station, fire_service, shelter, and admins) are designed with proper primary keys, data types, and relationships to ensure data consistency and reliability.

Additional Tools & Libraries :

XAMPP- XAMPP was used as the complete local development environment. It provided an easy-to-install Apache server, MySQL database, and phpMyAdmin in a single package, allowing seamless local hosting, database management, and testing of the entire system during development.

CORS (Cross-Origin Resource Sharing)- Implemented via the cors npm package to securely allow the frontend (running on the browser) to make API requests to the backend server, preventing cross-origin security issues while maintaining controlled access.

JSON Middleware- Built-in Express middleware (express.json()) was used to automatically parse incoming JSON request bodies, enabling smooth handling of data sent from frontend forms and fetch requests.

5. System Features :

Ambulance Services : The system allows users to search for ambulances based on their selected area. Users can view important details such as availability status, ambulance type, driver name, and the hospital linked to each ambulance service. Additionally, new ambulance service records can be added through the registration system.

Blood Bank & Donor Management : Users can search for blood banks by selecting their area. The system displays registered blood donors along with their blood group and current availability status. New donor information can also be added through a dedicated registration form.

Hospital & Doctor Information : The platform enables users to search for hospitals based on their selected area. Users can view complete hospital contact details including name, address, and phone number. For each hospital, the system fetches and displays associated doctors along with their specialty and availability status.

Police Services : Police services are searchable through a step-by-step process starting from district selection, followed by city, and finally area. Users can view detailed information about each police station including contact details and the name of the officer in charge.

Fire Service : Fire service stations can be searched based on district and area selection. The system provides complete information including the officer in charge, contact number, and full address of each fire station.

Emergency Shelters : Users can search for emergency shelters by selecting their area. The system displays important details such as the shelter name and its precise location.

System Architecture : The system follows a client-server architecture. On the client side, the web browser uses JavaScript to send requests to the server. The server side is powered by Express.js which handles all REST API requests. The database layer uses MySQL for persistent storage of all emergency service data.

System Flow : The user first selects the desired service category such as ambulance, hospital, blood bank, and others. Next, the user selects the appropriate location parameters including district, city, and area. The frontend then sends the request to the backend API. The backend queries the MySQL database to fetch the relevant data. Finally, the data is returned to the frontend and displayed dynamically to the user.

Backend Functionalities : The backend provides RESTful API endpoints for all emergency services to handle data retrieval and management. It uses secure database queries with parameterized SQL to prevent injection attacks. Dedicated data insertion APIs are available specifically for adding new ambulances and blood donor records. Comprehensive error handling and input validation are implemented for all incoming requests.

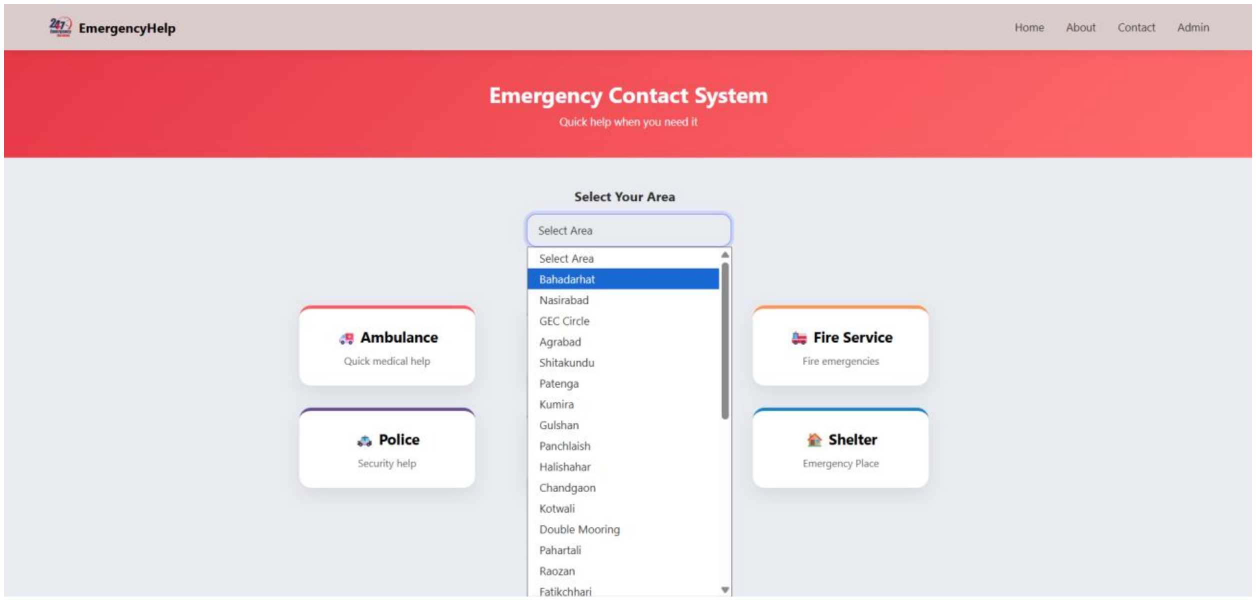
Database Interaction : The backend communicates with MySQL using structured SQL queries to perform various operations. It retrieves emergency service information from the database as needed. The system also inserts new records for ambulances and blood donors when registration forms are submitted. All data filtering is done efficiently based on area, district, or hospital as required by the user.

6. System Flow :

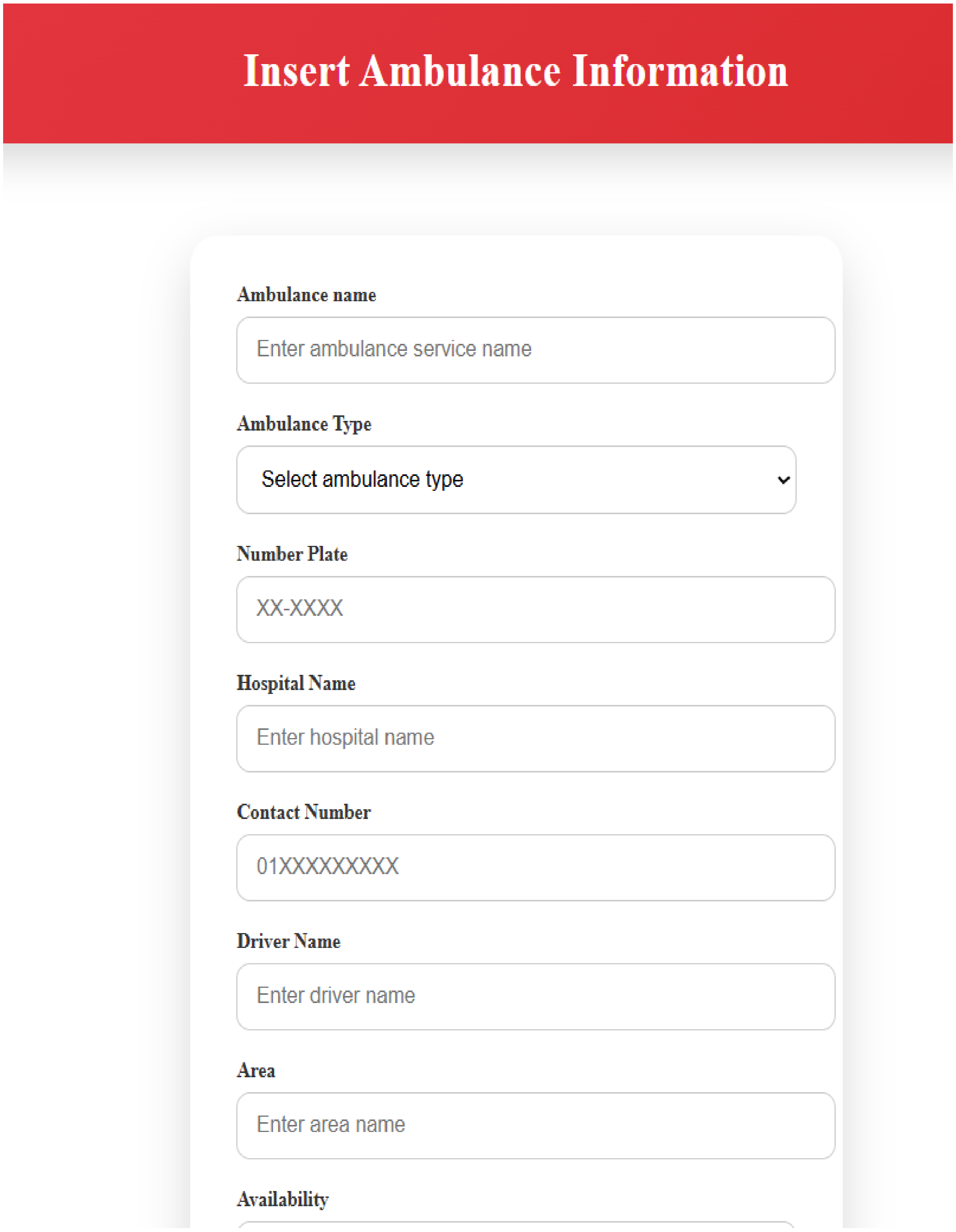
The Platform operates according to the following flow :

Select Your Area:

Users select their area to view location-based emergency services and resources relevant to their location.



Ambulance Insert for Users:
Users can submit ambulance service information through a form, which is stored in the system for public access.


Insert Ambulance Information

Ambulance name

Enter ambulance service name

Ambulance Type

Select ambulance type

Number Plate

XX-XXXX

Hospital Name

Enter hospital name

Contact Number

01XXXXXXXXX

Driver Name

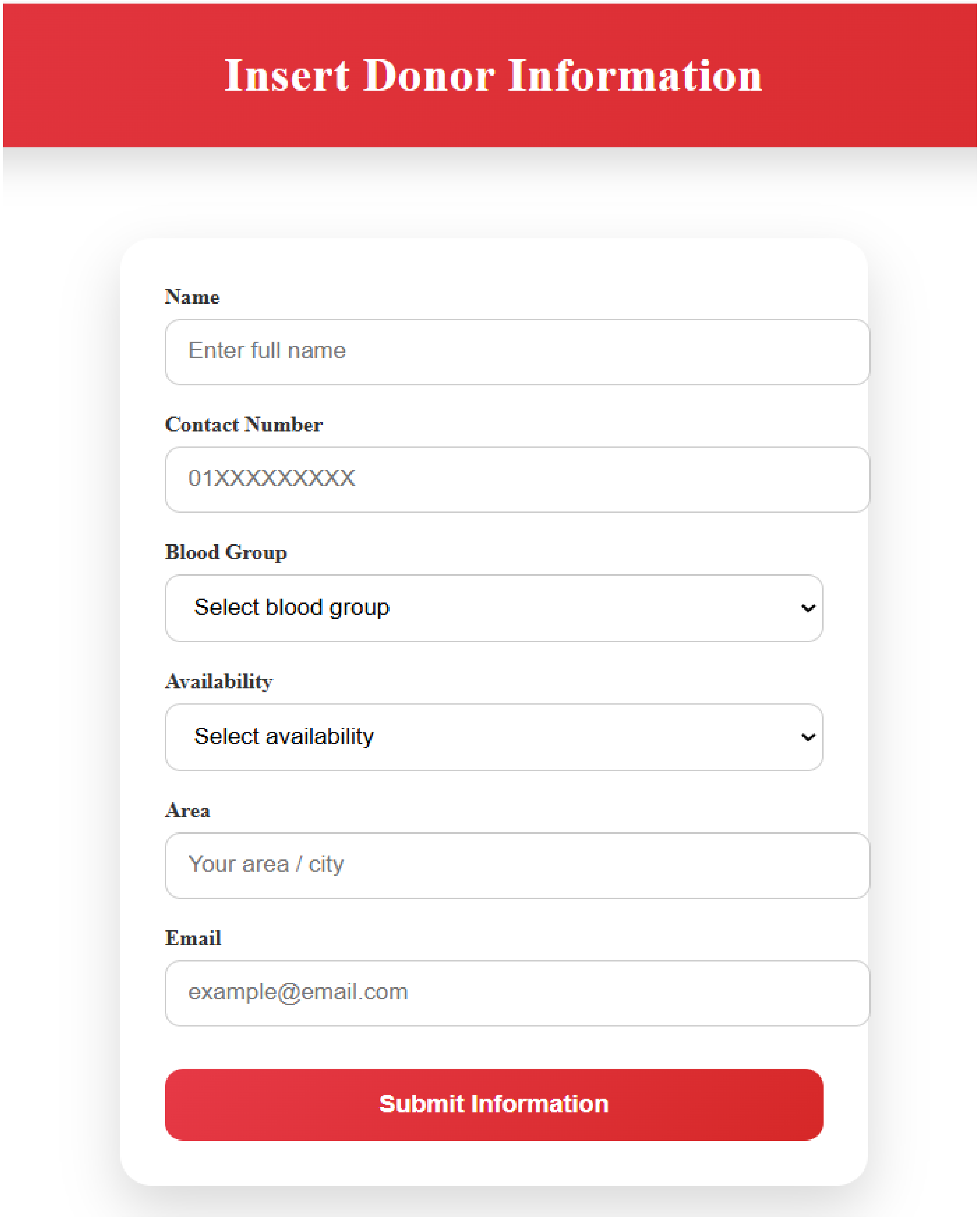
Enter driver name

Area

Enter area name

Availability

Donor Information Insert for Users:
Users can add blood donor details, including blood group, contact information, and availability, to help others during emergencies.


Insert Donor Information

Name

Enter full name

Contact Number

01XXXXXXXXX

Blood Group

Select blood group

Availability

Select availability

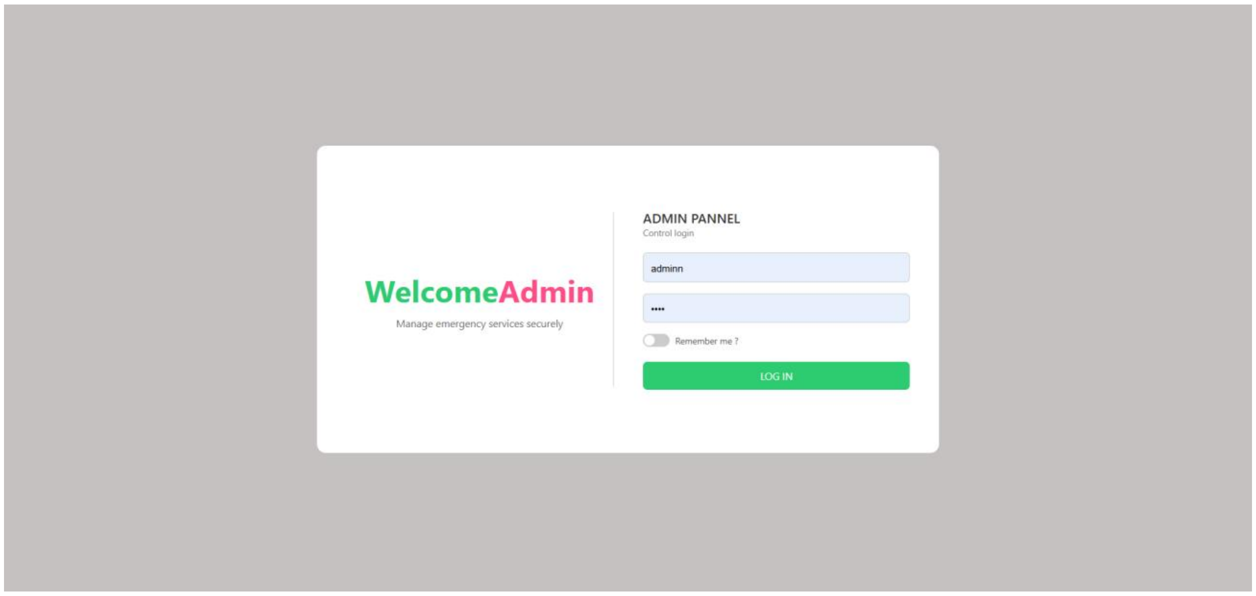
Area

Your area / city

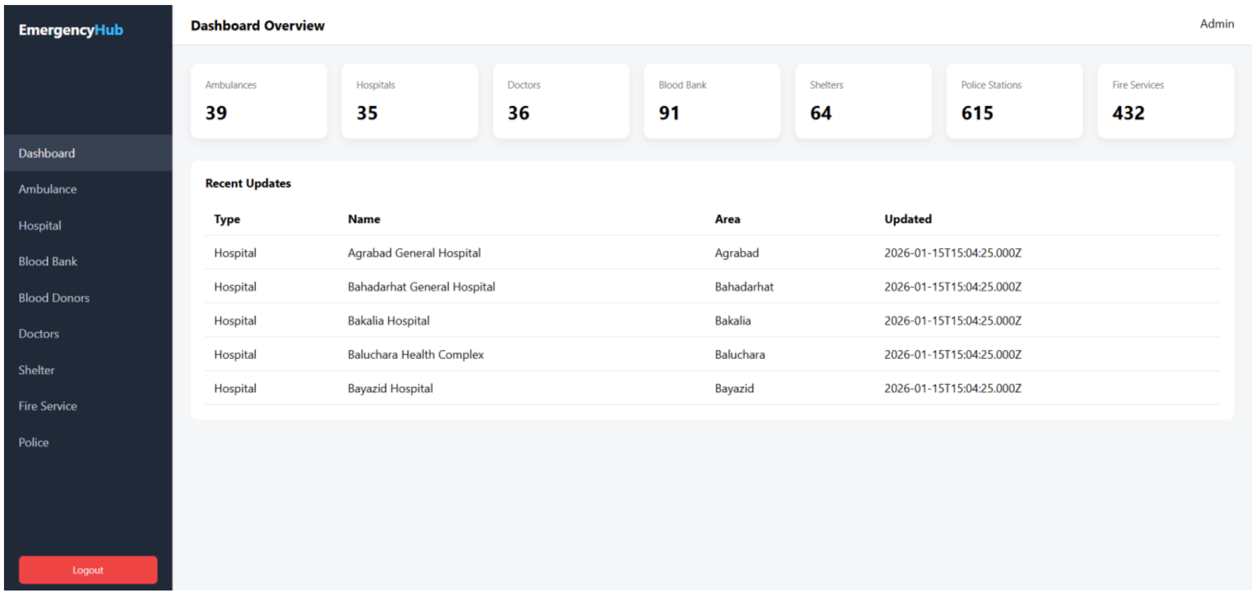
Email

example@email.com

Admin login Interface :
The admin login interface allows administrators to securely access the system by entering valid credentials to manage emergency service data.



Admin Dashboard – Full Access for Insert and Delete:
The admin dashboard provides administrators with full control to insert, update, and delete all emergency service records to maintain data accuracy.

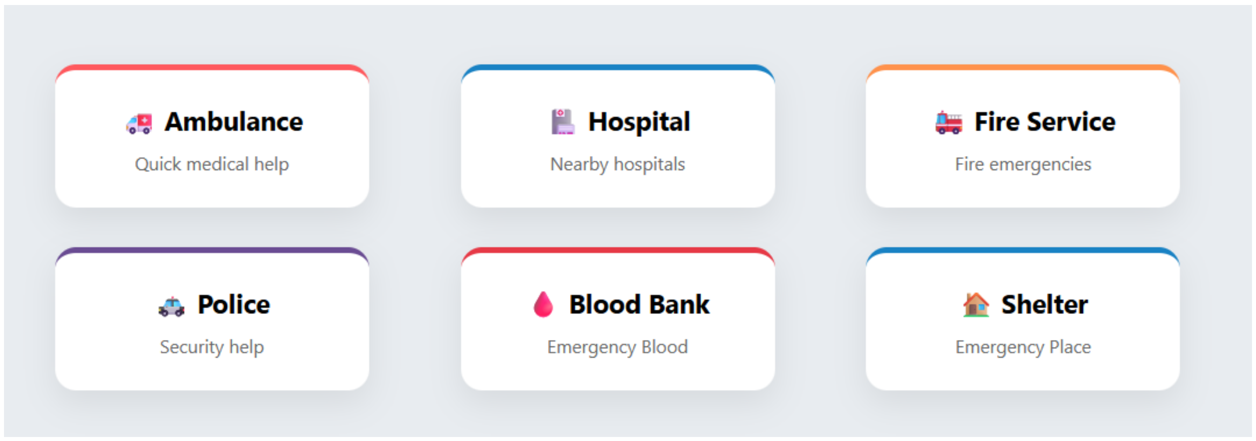


Ambulances	Hospitals	Doctors	Blood Bank	Shelters	Police Stations	Fire Services
39	35	36	91	64	615	432

Recent Updates

Type	Name	Area	Updated
Hospital	Agrabad General Hospital	Agrabad	2026-01-15T15:04:25.000Z
Hospital	Bahadarhat General Hospital	Bahadarhat	2026-01-15T15:04:25.000Z
Hospital	Bakalia Hospital	Bakalia	2026-01-15T15:04:25.000Z
Hospital	Baluchara Health Complex	Baluchara	2026-01-15T15:04:25.000Z
Hospital	Bayazid Hospital	Bayazid	2026-01-15T15:04:25.000Z

User Service Viewing Flow:
The user opens the website, selects an area from the dropdown list, and clicks on the desired service category such as Ambulance for quick medical help, Hospital to find nearby hospitals and available doctors, Fire Service for fire emergency assistance, Police for security and law enforcement support, Blood Bank to search for emergency blood and available donors, or Shelter to locate nearby safe shelters. Based on the selected area and service, the system queries the database and displays a list of available resources with important details such as contact numbers, addresses, and availability status, allowing users to view additional information when required.

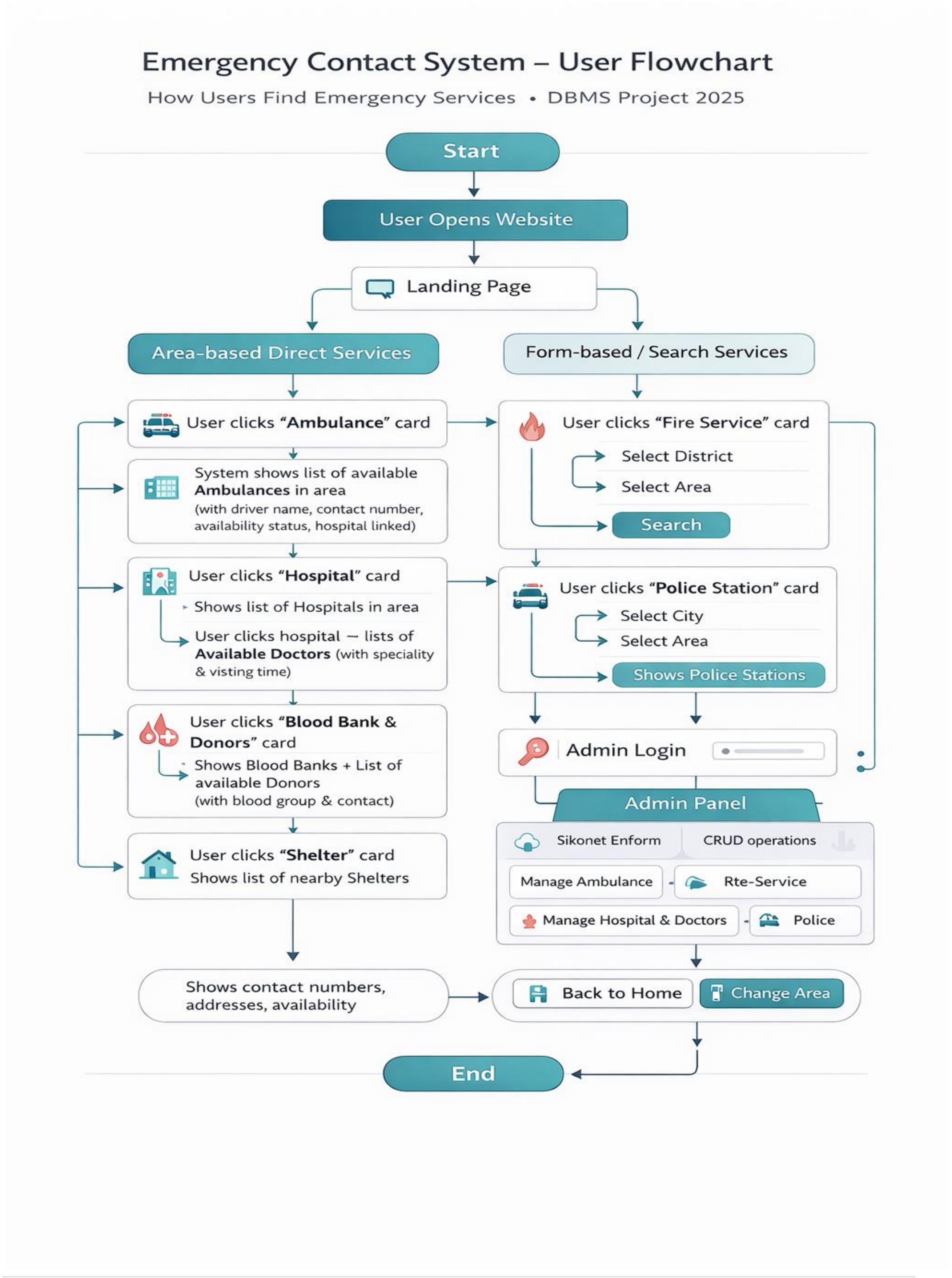


Flowchart :

The flowchart illustrates the overall working process of the Emergency Service Management System. It begins with the user opening the website and selecting an area to view location-based emergency services. The user then chooses a specific service category such as ambulance, hospital, blood bank, police, fire service, or shelter. Based on the selection, the system retrieves relevant data from the database and displays available services with important details. Users can view additional information like doctor availability or service status when required.

The flowchart also shows the registration process, where users can submit ambulance and blood donor information through input forms. Submitted data is sent to the backend and stored in the database for further use. Additionally, the admin flow is represented, where the admin logs in using valid credentials and accesses the admin dashboard. From the dashboard, the admin can insert, update, and delete emergency service records. The flowchart concludes by showing how the system ensures accurate data management and quick access to emergency services.

The flowchart shows how the frontend, backend, and database interact to process user requests efficiently and provide quick access to emergency information.

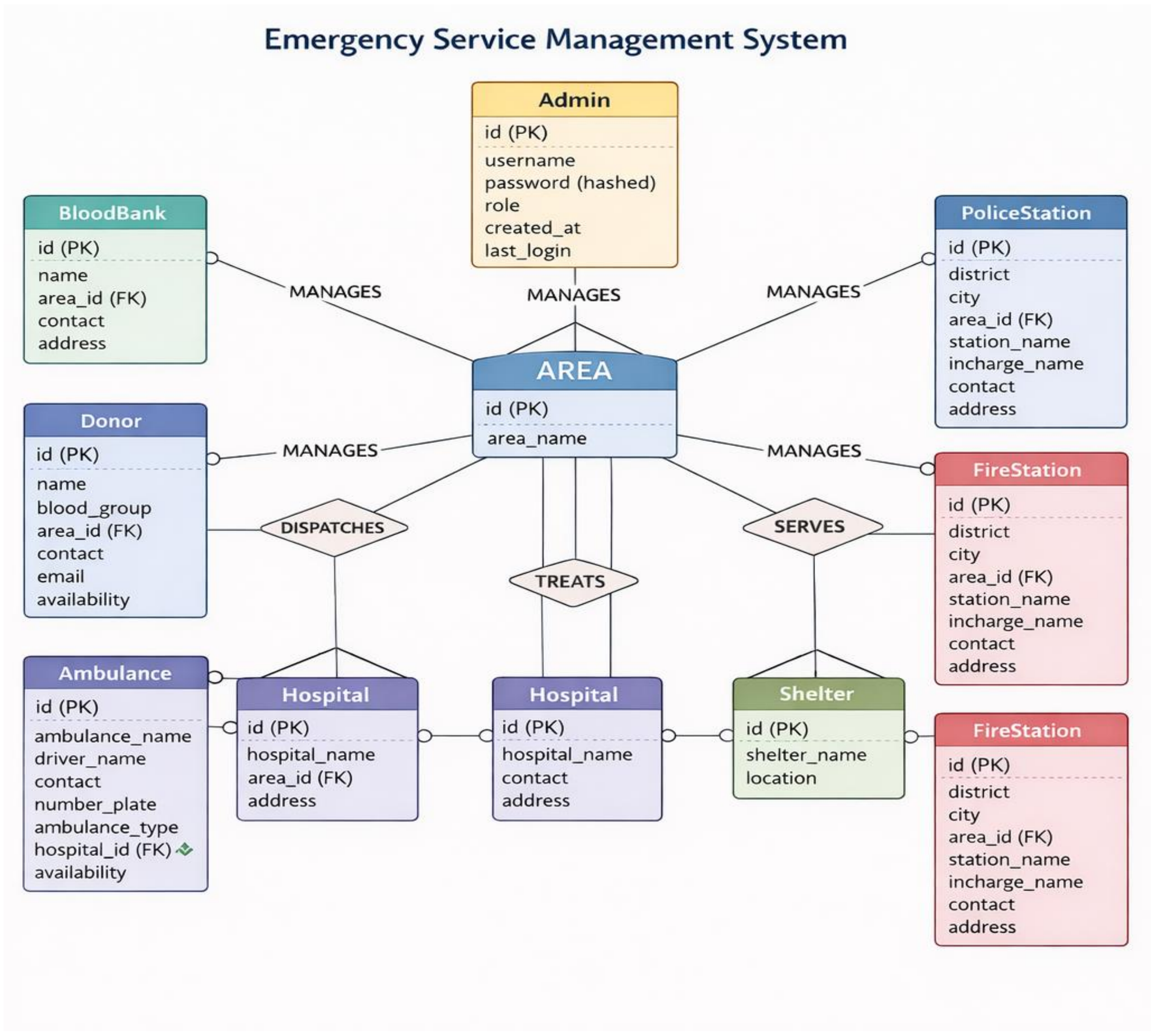

Emergency Contact System – User Flowchart
How Users Find Emergency Services • DBMS Project 2025
Start
User Opens Website
Landing Page
Area-based Direct Services
Form-based / Search Services
User clicks "Ambulance" card
System shows list of available Ambulances in area (with driver name, contact number, availability status, hospital linked)
User clicks "Hospital" card
Shows list of Hospitals in area
User clicks hospital – lists of Available Doctors (with speciality & visting time)
User clicks "Blood Bank & Donors" card
Shows Blood Banks + List of available Donors (with blood group & contact)
User clicks "Shelter" card
Shows list of nearby Shelters
User clicks "Fire Service" card
Select District
Select Area
Search
User clicks "Police Station" card
Select City
Select Area
Shows Police Stations
Admin Login
Admin Panel
Sikonet Enform
CRUD operations
Manage Ambulance
Rte-Service
Manage Hospital & Doctors
Police
Shows contact numbers, addresses, availability
Back to Home
Change Area
End

7. ER Diagram :

The database consists of multiple related tables including:

1. ambulance
2. hospital
3. doctor
4. bloodbank
5. personinformation (donors)
6. police_station
7. fire_service
8. shelter

Each table contains a primary key and relevant attributes. Foreign key relationships are used where necessary, such as doctors linked to hospitals.



8. Database Interaction :

The system is connected to a MySQL database running on the local host environment to store and retrieve emergency service information. Backend code establishes a secure connection with the database and executes queries to fetch, insert, and manage data in real time. phpMyAdmin is used to monitor the database structure, tables, and records during development.

```
   const express = require('express');
const mysql = require('mysql');
const cors = require('cors');

const app = express();
app.use(cors());
app.use(express.json());


const path = require("path");
app.use(express.static(path.join(__dirname)));



// MySQL connection
const db = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '',
    database: 'emergency_db'
});

db.connect(err => {
    if (err) {
        console.error('Database connection failed:', err);
        return;
    }
    console.log('Connected to MySQL');
});
```

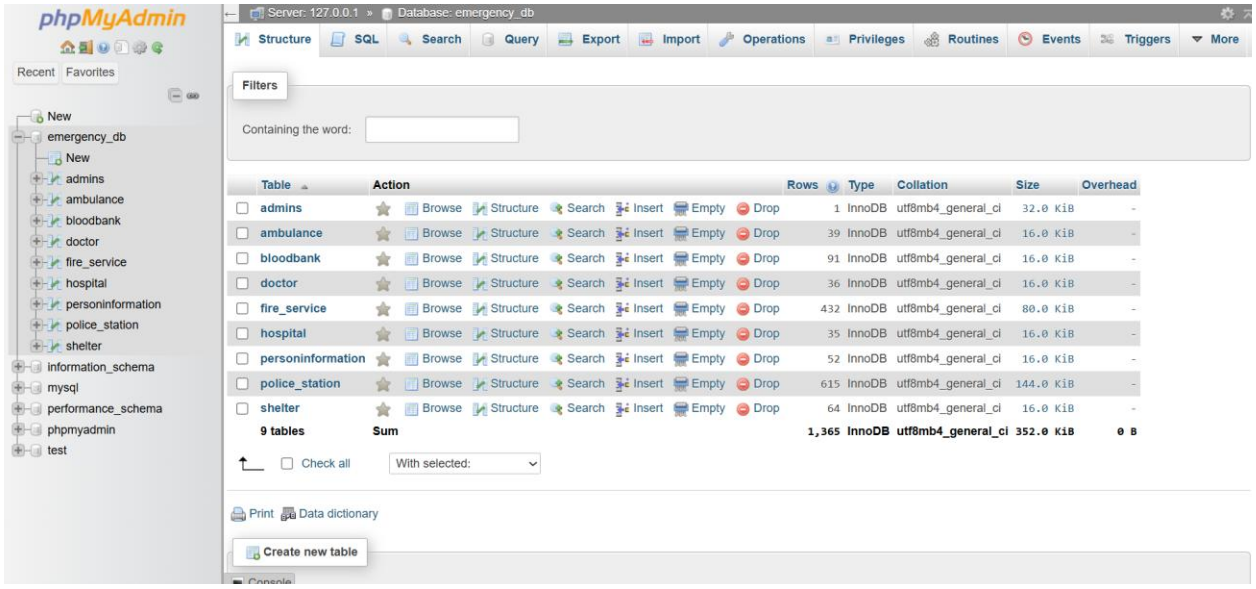
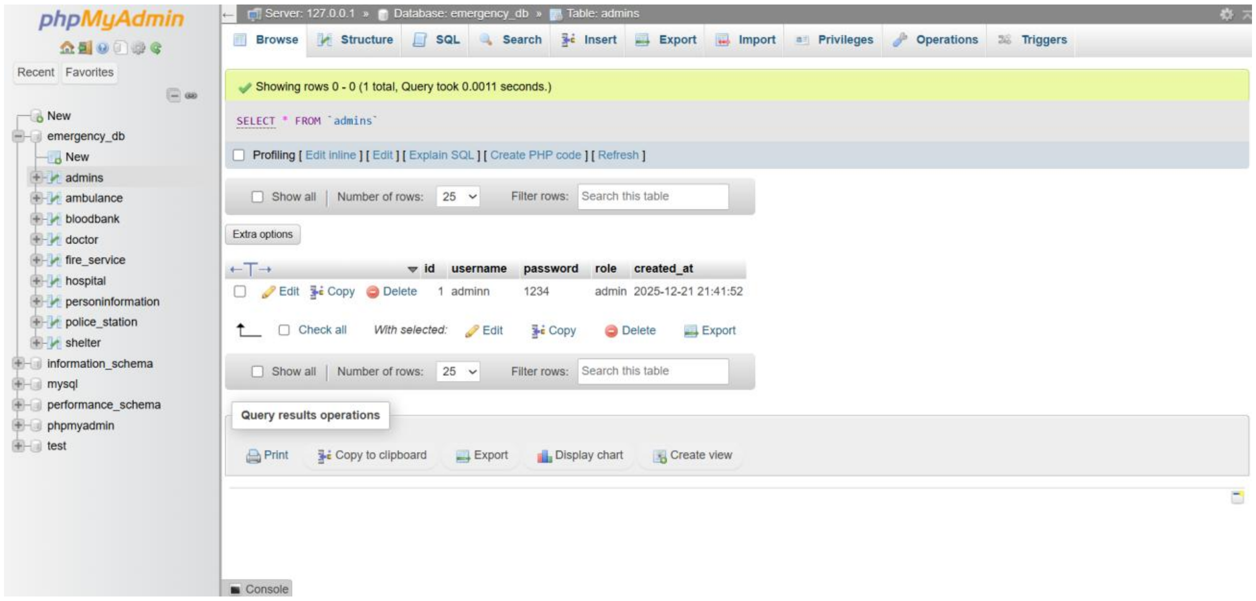


Table	Action	Rows	Type	Collation	Size	Overhead
admins	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
ambulance	Browse Structure Search Insert Empty Drop	39	InnoDB	utf8mb4_general_ci	16.0 KiB	-
bloodbank	Browse Structure Search Insert Empty Drop	91	InnoDB	utf8mb4_general_ci	16.0 KiB	-
doctor	Browse Structure Search Insert Empty Drop	36	InnoDB	utf8mb4_general_ci	16.0 KiB	-
fire_service	Browse Structure Search Insert Empty Drop	432	InnoDB	utf8mb4_general_ci	80.0 KiB	-
hospital	Browse Structure Search Insert Empty Drop	35	InnoDB	utf8mb4_general_ci	16.0 KiB	-
personinformation	Browse Structure Search Insert Empty Drop	52	InnoDB	utf8mb4_general_ci	16.0 KiB	-
police_station	Browse Structure Search Insert Empty Drop	615	InnoDB	utf8mb4_general_ci	144.0 KiB	-
shelter	Browse Structure Search Insert Empty Drop	64	InnoDB	utf8mb4_general_ci	16.0 KiB	-
9 tables	Sum	1,365	InnoDB	utf8mb4_general_ci	352.0 KiB	0 B

9. Admin Panel Interaction :

The admin panel is accessed through the local host environment, where administrators can securely manage emergency service data. From this panel, the admin can insert, update, and delete records to ensure the system remains accurate and up to date.


phpMyAdmin
Recent Favorites
New
emergency_db
New
admins
ambulance
bloodbank
doctor
fire_service
hospital
personinformation
police_station
shelter
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: emergency_db » Table: admins
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print Copy to clipboard Export Display chart Create view
Console

Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)

```
SELECT * FROM `admins`
```

	id	username	password	role	created_at
Edit Copy Delete	1	adminn	1234	admin	2025-12-21 21:41:52

10. Challenges Faced :

Integrating Real-Time Availability:

Ensuring that real-time availability of ambulances, doctors, and blood donors was accurately reflected in the system proved to be challenging. This required efficient MySQL queries and proper backend logic using Node.js and Express to ensure that availability status was updated and retrieved correctly without performance issues.

Managing Location-Based Data Filtering:

Handling district, city, and area-based filtering for services such as police stations, fire services, hospitals, and shelters required careful database design and optimized queries. Ensuring accurate and fast results based on user-selected locations was a key technical challenge.

User Interface Design:

Designing a clean and user-friendly interface that allows users to easily search emergency services during critical situations was challenging. Special attention was required to maintain simplicity, readability, and quick navigation while presenting large amounts of emergency-related information.

Admin Panel Functionality:

Developing an effective admin panel that provides full control over managing emergency service records was crucial. Ensuring that administrators could efficiently add, update, and monitor ambulance services, donors, hospitals, and other emergency resources posed both design and functional challenges.

Frontend and Backend Synchronization:

Maintaining smooth communication between the frontend JavaScript and backend REST APIs was challenging. Proper error handling, data validation, and API response management were implemented to ensure reliable data exchange across the system.

11. Future Enhancements :

Automatic detection of user location using browser Geolocation API One-click calling feature to emergency numbers SMS/Email notification system for critical requests User rating and feedback system for services Separate dashboard for service providers to self-update information Multi-language support (Bangla + English) Mobile application version (Android/iOS).

Conclusion :

The Emergency Service Management System provides a centralized platform for quick access to critical emergency services, improving response time and public safety. With future enhancements, it has the potential to become a comprehensive emergency response solution for cities and communities.

Team Members and Contributions :

Team Leader:
Mohammad Sadman Tahiat **(ID: C241100)**

Contribution: 35%
Led the overall development of the Emergency Service Management System with a primary focus on backend implementation using Node.js and Express.js. Managed database design and integration with MySQL, developed RESTful APIs for emergency services, and ensured smooth data flow between frontend and backend. Contributed to minimal frontend development for data display and user interaction, assisted in designing the admin panel structure, and coordinated team activities, testing, and deployment.

Team Member 1:
Minhaj Hasan Rohan **(ID: C241101)**

Contribution: 35%
Played a significant role in backend development by assisting in API implementation, database interaction using MySQL, and integrating backend logic with the system features. Contributed to connecting RESTful APIs with frontend components and handling data flow between client and server. Also provided minimal frontend support using HTML, CSS, and JavaScript, and assisted in designing and improving the admin panel functionality and usability.

Team Member 2:
Ahsan Miran **(ID: C241099)**

Contribution: 30%
Contributed to frontend development by implementing service listing pages, dynamic content rendering, and user interaction features using JavaScript. Assisted in improving UI consistency and layout design across different modules such as hospitals, blood banks, and shelters. Supported testing and debugging of frontend functionality to ensure smooth integration with backend services.